The Short Introduction to Imbalanced Classification

Zeyu Qin

07.02.2020

Overview

Reference

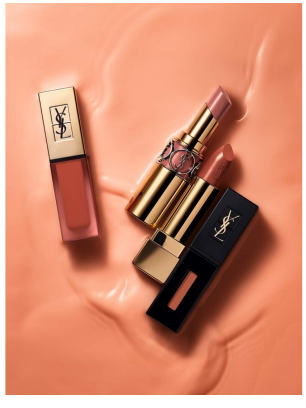
- Cui, Yin, et al. "Class-balanced loss based on effective number of samples." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019. [1]
- Cao, Kaidi, et al. "Learning imbalanced datasets with label-distribution-aware margin loss." Advances in Neural Information Processing Systems. 2019. [2]
- Kang, Bingyi, et al. "Decoupling representation and classifier for long-tailed recognition." arXiv preprint arXiv:1910.09217 (2019).
- Chatterjee, Satrajit. "Coherent Gradients: An Approach to Understanding Generalization in Gradient Descent-based Optimization." arXiv preprint arXiv:2002.10657 (2020).

Learning from Imbalanced Data

The necessity of Imbalanced Classification (IC)

Disease diagnosis based on medical records: For example, suppose you are building a model which will look at a person's medical records and classify whether or not they are likely to have a rare disease. An accuracy of 99.5% might look great until you realize that it is correctly classifying the 99.5% of healthy people as "disease-free" and incorrectly classifying the 0.5% of people which do have the disease as healthy.

Non-IID in Distributed optimization and Federated Learning: In some extreme cases, there are only one or two classes of data in each client.




The reason of why IC could affect the ML model

If we're updating a parameterized model by gradient descent to minimize our loss function, we'll be spending most of our updates changing the parameter values in the direction which allow for correct classification of the majority class.

In other words, many machine learning models are subject to a **frequency bias** in which they place more emphasis on learning from data observations which occur more commonly.

It's worth noting that not all datasets are affected equally by class imbalance. Generally, for easy classification problems in which there's a clear separation in the data, class imbalance doesn't impede on the model's ability to learn effectively.

Short and Simple Explanations

A simple but non-trivial example:
for the simple linear model with soft-max classifier or the last classifier of the Deep neural net. **x** is our input or the feature from DNN and **K** is the class number. So the training output is

$$
\begin{aligned}
s_i =& \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad z_i = w_i' x + b_i \\
L =& \sum_{i=1}^{K} -y_i \ln s_i
\end{aligned}
\tag{1}
$$

Let's dive deeper, do some simple gradient computation for CE loss.

$$
\begin{aligned}
\frac{\partial L}{\partial w_i} =& (s_i - y_i)x \\
\frac{\partial L}{\partial b_i} =& s_i - y_i
\end{aligned}
\tag{2}
$$

Short and Simple Explanations

So from the above equations, we can see the $\frac{\partial L}{\partial w_i}$ has the reverse direction with x because of $s_i \leq y_i$. So, if we run the SGD and some variants, the search direction is always the same as the sample x. **So, for the supervised learning, the similar samples (with the same label) have the similar gradients, especially the linear separable data.**

Classical Methods for Imbalanced Classification

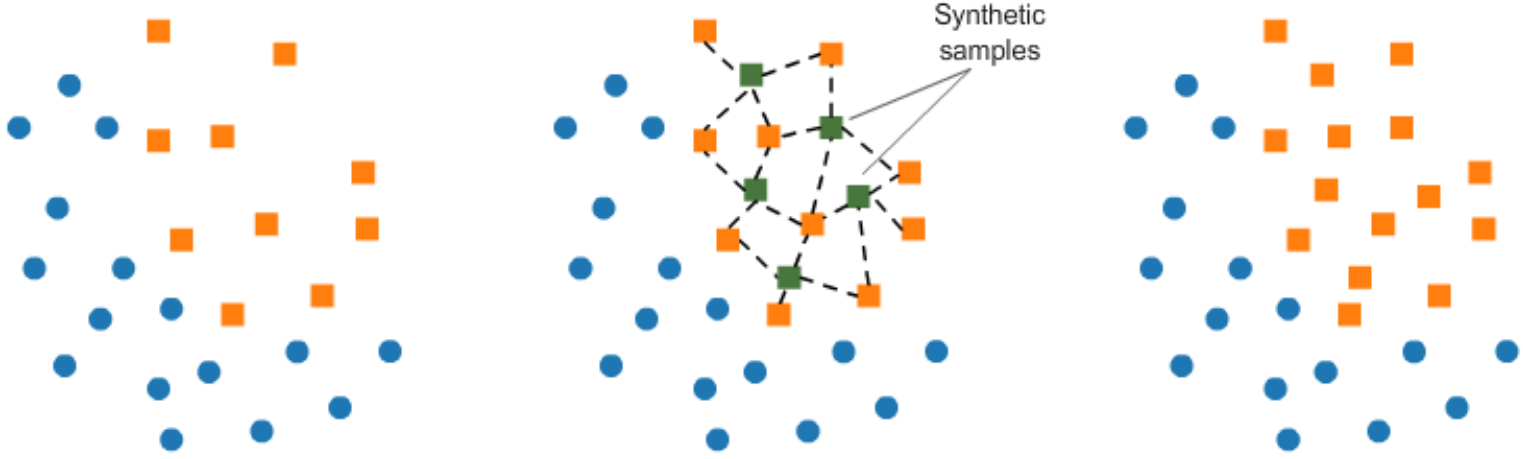
Classical methods

- Re-sampling: Over-sampling, Up-sampling
- Re-weighting: Cost-sensitive loss

These two mainstreamed methods is designed to solve the frequency bias for large-scale machine learning.

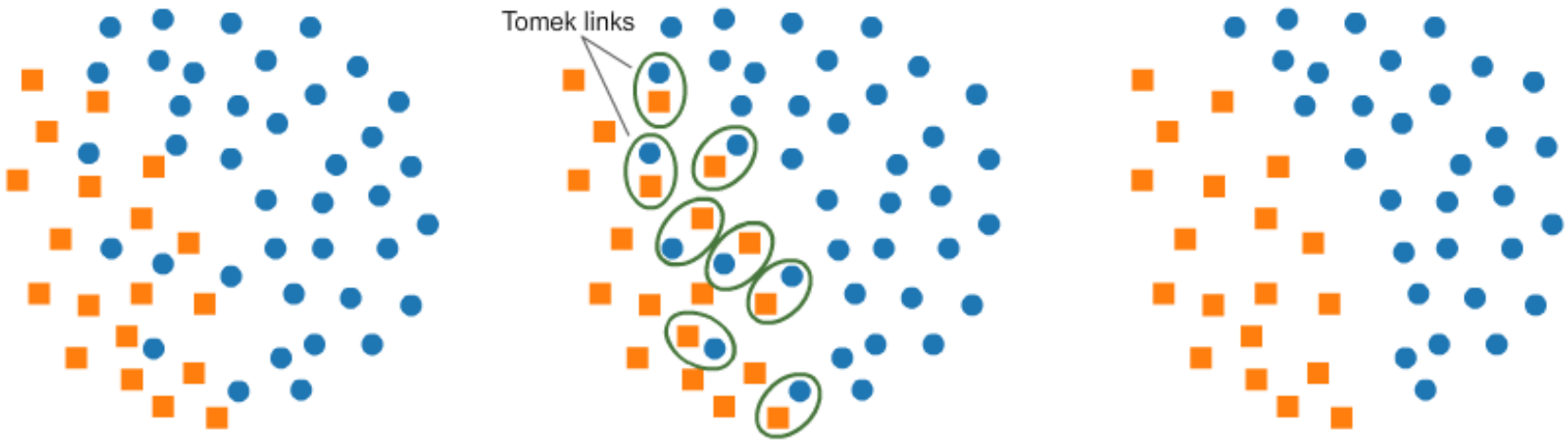
Re-sampling

Re-sampling. There are two types of re-sampling techniques: over-sampling the minority classes and under-sampling the frequent classes.
Over-sampling: Over-Sampling increases the number of instances in the minority class by randomly replicating them. Rather than simply replication, we could use **data augmentation** and synthetic instances by **interpolation**.



Re-sampling

Up-sampling:Up-sampling essentially throws away data from major class to make it easier to learn characteristics about the minority classes. it will simply "clean" the dataset by removing some noisy observations, which may result in an easier classification problem. (margin and stability of model)



Re-weighting

Re-weighting: (Cost-sensitive) this method is almost same as Over-sampling. Cost-sensitive re-weighting assigns (adaptive) weights for different classes or even different samples.

We want to place more emphasis on the minority classes such that the end result is a classifier which can learn equally from all classes.

Weighting by **inverse class frequency or a smoothed version of inverse square root of class frequency** are often adopted.

$$\mathrm{CB}(\mathbf{p}, y) = \alpha_i \mathcal{L}(\mathbf{p}, y), \text{for each class}$$

Problems:
(a) Re-sampling the examples in minority classes often causes heavy over-fitting to the minority classes when the model is a deep neural network, as pointed out in prior work

(b) weighting up the minority classes' losses can cause difficulties and instability in optimization, especially when the classes are extremely imbalanced

The advanced methods used for DNN

Effective Number of the Class

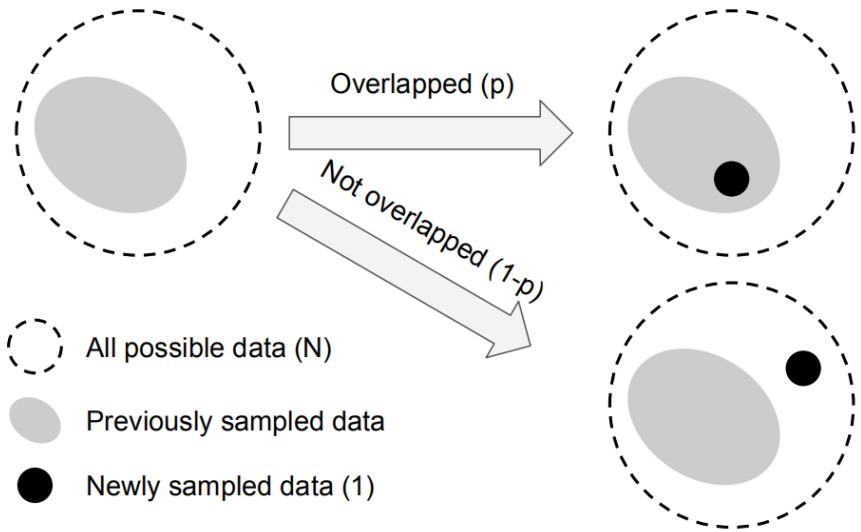
This method focuses on choosing the weight of cost function.
The important question: can the sample number of each class define the size of class?

- as we know, the data from the same class share lots of similarities. So, lots of data always stay in a small neighboring region in the feature space.
- Data from the same class can always be represented by some typical samples that we call **prototypes**.
- The prototypes have the larger effect on the optimization process.

Effective Number of the Class

The effective number of samples is the expected volume of samples, but is very difficult to compute because it depends on the shape of the sample and the dimensionality of the feature space. Here, we only consider two cases: entirely inside the set of previously sampled data or entirely outside.

Given a class, denote the set of all possible data in the feature space of this class as $\mathcal{S}$. We assume the volume of $\mathcal{S}$ is N and $N \geq 1$. Denote each data as a subset of $\mathcal{S}$ that has the unit volume of 1 and may overlap with other data. We denote the effective number of sample as E_n.



Effective number of the Class

Proposition

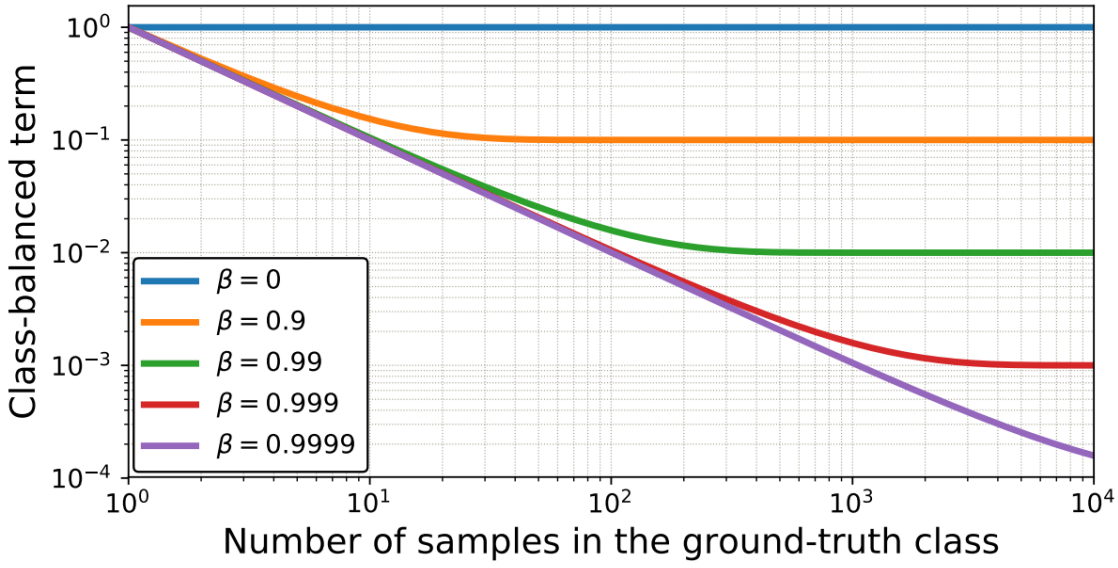
Effective Number. $E_n = (1-\beta^n)/(1-\beta)$ *where* $\beta = (N-1)/N$.

Proof: We can prove it by using the first induction method.

Implication(Asymptotic Properties) $E_n = 1$ if $\beta = 0$ $(N = 1)$.$E_n \to n$ as $\beta \to 1 (N \to \infty)$
Proof: We can prove it by using the L'Hopital's rule.

The asymptotic property of E_n shows that when N is large, the effective number of samples is same as the number of samples n. In practice, we assume Ni is only dataset-dependent and set $N_i = N, \beta_i = \beta = (N-1)/N$ for all classes in a dataset. Actually we only determine the β.

Effective number of the Class



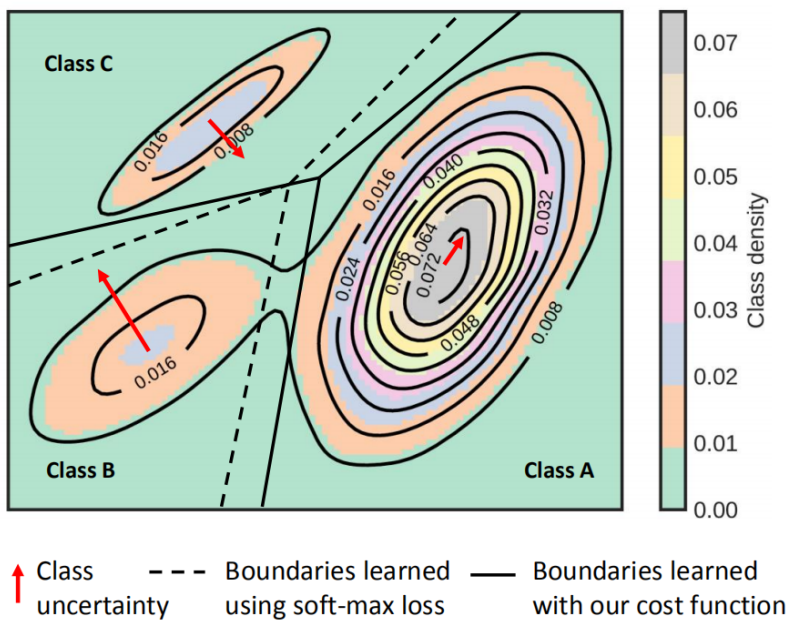
Dataset Name	Long-Tailed CIFAR-10						Long-Tailed CIFAR-100					
Imbalance	200	100	50	20	10	1	200	100	50	20	10	1
Softmax	**34.32**	29.64	25.19	17.77	13.61	6.61	65.16	61.68	56.15	48.86	44.29	29.07
Sigmoid	34.51	**29.55**	23.84	**16.40**	**12.97**	**6.36**	64.39	**61.22**	55.85	48.57	44.73	**28.39**
Focal ($\gamma = 0.5$)	36.00	29.77	**23.28**	17.11	13.19	6.75	65.00	61.31	55.88	48.90	44.30	28.55
Focal ($\gamma = 1.0$)	34.71	29.62	23.29	17.24	13.34	6.60	**64.38**	61.59	**55.68**	**48.05**	**44.22**	28.85
Focal ($\gamma = 2.0$)	35.12	30.41	23.48	16.77	13.68	6.61	65.25	61.61	56.30	48.98	45.00	28.52
Class-Balanced	**31.11**	**25.43**	**20.73**	**15.64**	**12.51**	**6.36***	**63.77**	**60.40**	**54.68**	**47.41**	**42.01**	**28.39***
Loss Type	SM	Focal	Focal	SM	SGM	SGM	Focal	Focal	SGM	Focal	Focal	SGM
β	0.9999	0.9999	0.9999	0.9999	0.9999	-	0.9	0.9	0.99	0.99	0.999	-
γ	-	1.0	2.0	-	-	-	1.0	1.0	-	0.5	0.5	-

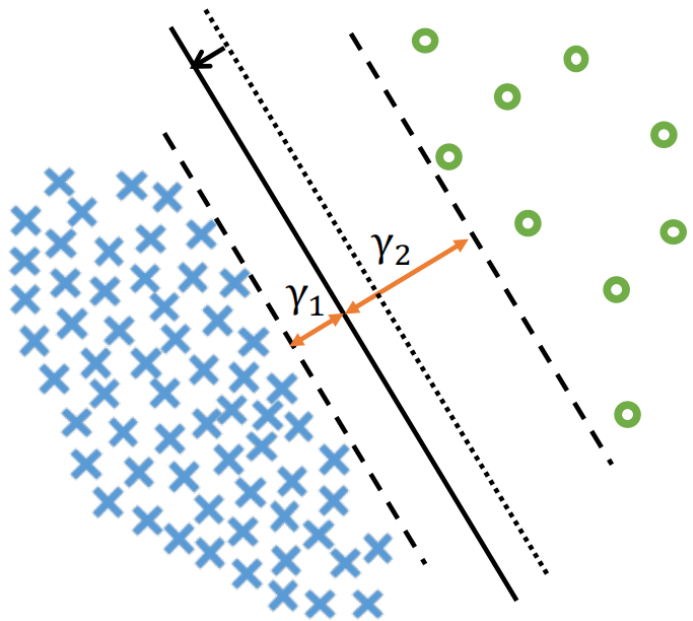
Label-Distribution-Aware Margin Loss (LDAM)

LDAM

This paper is based on the classical notion — margin which is also associated with stability.

A consensus in ML and over-parameterized DNN is the model with larger margin has the better generalization. This paper[1] also proves that over-parameterized DNN converges the max-margin solution with SGD.





[1] Soudry, Daniel, et al. "The implicit bias of gradient descent on separable data." The Journal of Machine Learning Research 19.1 (2018): 2822-2878.

LDAM

Regularizing the minority classes more strongly than the frequent classes so that we can improve the generalization error of minority classes without sacrificing the model's ability to fit the frequent classes.

Define the training margin for class j as:

$\gamma_j = \min_{i \in S_j} \gamma(x_i, y_i)$, $\gamma_{\min} = \min\{\gamma_1, \ldots, \gamma_k\}$ The typical generalization error bounds scale in $\mathrm{C}(\mathcal{F})/\sqrt{n}$. That is, in our case, if the test distribution is also imbalanced as the training distribution, then

$$\text{imbalanced test error} \lesssim \frac{1}{\gamma_{\min}} \sqrt{\frac{\mathrm{C}(\mathcal{F})}{n}}$$

Theorem

With high probability $\left(1 - n^{-5}\right)$ *over the randomness of the training data, for the balanced test data, we have the generalization bound:*

$$L_{\mathrm{bal}}[f] \lesssim \frac{1}{k} \sum_{j=1}^{k} \left(\frac{1}{\gamma_j} \sqrt{\frac{\mathrm{C}(\mathcal{F})}{n_j}} + \frac{\log n}{\sqrt{n_j}} \right)$$

LDAM

How to determine the γ?

For the simple binary classification problem,

$$\min_{\gamma_1+\gamma_2=\beta} \frac{1}{\gamma_1}\sqrt{\frac{1}{n_1}} + \frac{1}{\gamma_2}\sqrt{\frac{1}{n_2}}$$

$$\frac{1}{(\beta-\gamma_1)^2\sqrt{n_2}} - \frac{1}{\gamma_1^2\sqrt{n_1}} = 0$$

So the solution is $\gamma_1 = \frac{C}{n_1^{1/4}}$, and $\gamma_2 = \frac{C}{n_2^{1/4}}$. And the solution for multiclass classification, the class-dependent margin is

$$\gamma_j = \frac{C}{n_j^{1/4}}$$

$$\mathcal{L}_{\mathrm{LDAM}}((x,y);f) = -\log \frac{e^{z_y-\gamma_y}}{e^{z_y-\gamma_y}+\sum_{j\neq y} e^{z_j}}$$

$$\textbf{where } \gamma_j = \frac{C}{n_j^{1/4}} \textbf{ for } j \in \{1,\ldots,k\}$$

LDAM

Two-stage training: Deferred Re-balancing Optimization Schedule, in the first stage, we only train our model with LDAM in imbalanced training dataset (no RW,RS). Then, in the second stage, we also use RW or RS.

- Top-1 validation errors on imbalanced IMDB review dataset

Approach	Error on positive reviews	Error on negative reviews	Mean Error
ERM	2.86	70.78	36.82
RS	7.12	45.88	26.50
RW	5.20	42.12	23.66
LDAM-DRW	4.91	30.77	17.84

- Top-1 validation errors of ResNet-32 on imbalanced CIFAR-10 and CIFAR-100.

Dataset	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	long-tailed		step		long-tailed		step	
Imbalance Ratio	100	10	100	10	100	10	100	10
ERM	29.64	13.61	36.70	17.50	61.68	44.30	61.45	45.37
Focal [Lin et al., 2017]	29.62	13.34	36.09	16.36	61.59	44.22	61.43	46.54
LDAM	26.65	13.04	33.42	15.00	60.40	43.09	60.42	43.73
CB RS	29.45	13.21	38.14	15.41	66.56	44.94	66.23	46.92
CB RW [Cui et al., 2019]	27.63	13.46	38.06	16.20	66.01	42.88	78.69	47.52
CB Focal [Cui et al., 2019]	25.43	12.90	39.73	16.54	63.98	42.01	80.24	49.98
HG-DRS	27.16	14.03	29.93	14.85	-	-	-	-
LDAM-HG-DRS	24.42	12.72	24.53	12.82	-	-	-	-
M-DRW	24.94	13.57	27.67	13.17	59.49	43.78	58.91	44.72
LDAM-DRW	**22.97**	**11.84**	**23.08**	**12.19**	**57.96**	**41.29**	**54.64**	**40.54**

LDAM

Strange knowledge emerges.

Loss	Schedule	Top-1	Top-5
ERM	SGD	42.86	21.31
CB Focal [Cui et al., 2019]	SGD	38.88	18.97
ERM	DRW	36.27	16.55
LDAM	SGD	35.42	16.48
LDAM	**DRW**	**32.00**	**14.82**